



“FINAL REPORT: A DATA ANALYTIC APPROACH ON WINE DATA”

IENG3304 – DATA MANAGEMENT & ANALYTICS



JUNE 21, 2022

WRITTEN BY: ABILASH SURENDRAN

BANNER ID: B00891410

TABLE OF CONTENT:

Serial No	Content	Page Number
1	TABLE OF FIGURES	1
2	INTRODUCTION	3
3	DATASET	3
4	DATA EXPLORATION AND VISUALISATION	5
5	CLUSTERING	7
6	REGRESSION MODEL	10
7	CLASSIFICATION I	12
8	CLASSIFICATION II	14
9	CONCLUSION	16
10	REFERENCE	16
11	APPENDIX A: DATASET	16
12	APPENDIX B: PRIOR PROBABILITIES	17
13	R CODE	19

TABLE OF FIGURES:

Serial No	FIGURES	Page Number
1	<i>Fig 1: Histogram on Fixed Acidity Level</i>	4
2	<i>Fig 2: Scatter Plot for Alcohol and pH</i>	5
3	<i>Fig 3: Box Plot Between Quality and Alcohol Level in Wines</i>	6
4	<i>Fig 4: Box Plot Between Quality and pH Level in Wines</i>	6
5	<i>Fig 5: Scatter Plot for Residual Sugar and Density</i>	7
6	<i>Fig 6: scatter plot for pH and Chlorides</i>	7
7	<i>Fig 7: $k = 3$, Clustering on Fixed Acidity of Wines</i>	8
8	<i>Fig 8: Scatter Plot for Residual Sugar and Density With $k = 3$</i>	8
9	<i>Fig 9: Scatter Plot for pH and chlorides With $k = 3$</i>	9
10	<i>Fig 10: Scatter Plot for pH and alcohol With $k = 3$</i>	9
11	<i>Fig 11: plot between r square value and model complexity of model 1</i>	10
12	<i>Fig 12: plot between r square value and model complexity of model 2</i>	10

13	<i>Fig 13: plot between r square value and model complexity of model 3</i>	11
14	<i>Fig 14: plot between r square value and model complexity of model 4</i>	11
15	<i>Fig 15: plot between r square value and model complexity of model 5</i>	11
16	<i>Fig 16: Regression curves between pH and alcohol level in wines</i>	11
17	<i>Fig 17: Plot between r square values and model complexity of the altered model 5.</i>	12
18	<i>Fig 18: 5-fold error bar graph for classification of wines based on quality</i>	13
19	<i>Fig 19: visualising error rates in classifying wines</i>	14
20	<i>Fig 20: Categorising wines based on different qualities</i>	14
21	<i>Fig 21: 5-fold error bar graph for classification of wines based on new categorised qualities</i>	15
22	<i>Fig 22: visualising error rates in classifying wines</i>	15
23	<i>Fig 23: visualising the actual number of wines classified as groups</i>	15
24	<i>Fig 24: Screenshot of the wine dataset with the actual quality</i>	16
25	<i>Fig 25: Screenshot of the wine dataset with categorised quality</i>	17

Introduction:

Wine is an alcoholic beverage made mostly from grapes that have been fermented. Wine has been produced for thousands of years. The Caucasus region, which is now Georgia, Persia, and Italy, was the first to develop wine. The red wine industry has undergone recent exponential growth as social drinking grows increasingly popular. Industry participants are now using product quality certificates to sell their items. This set of data pertains to Portuguese red "Vinho Verde" wine variants. Vinho Verde is a refreshing summer wine with tart acidity, low alcohol, and a hint of fizz. Vinho Verde wines are prepared from a blend of native Portuguese grapes and are not matured before being released. The grape varieties amarel, azal tinto, borraçal, brancelho, espadeiro, padeiro, pedral, rabo de ovelha, and vinho are recommended for red Vinho Verde [2]. Vinho is the most widely planted red grape variety, producing low-alcohol, structured wines with herbal, peppery notes. In this report, we will look at various data visualisation techniques that are used to visualise the wine dataset. Also, regression and classification techniques are used to establish the relationship between several dependent and independent variables within the dataset and to classify the wines based on their quality.

DATASET:

This set of data pertains to red "Vinho Verde" wine variants from Portugal. The dataset consists of 1143 rows and 12 columns of data. Some of the columns are alcohol, chlorides, citric acid, density, fixed acidity, free sulphur dioxide, Id, pH, quality, residual sugar, sulphates, total sulphur dioxide, and volatile acidity. This dataset helps in performing regression, classification and clustering. The data reveals how many active components are present in wine and how they affect the quality of the wine. The data is clustered using the datasets. The study concentrates on how these chemical qualities influence the quality of the wine. As minute changes in the chemical composition of wine occur, the quality of the wine tends to shift. Let's have a look at each parameter individually:

Alcohol:

The alcohol content of the wine dataset is shown in this column. The amount of alcohol in a wine influences its flavour and texture, as well as how much alcohol evaporates to transport the wine's smell to our senses. Alcohol adds viscosity, which helps to balance sweetness and acidity.

Volatile acidity:

This column is to account for the high acetic acid content in wines, which gives them a vinegary flavour. In general, the maximum volatile acidity for red wines should be around 0.14 g/100mL. The wine has a harsh, vinegary tactile sensation as the volatile acidity rises, which is generated by the acetic acid.

Sulphates:

Yeast is employed to ferment the wine, which results in the production of sulphites. Since the 1800s, winemakers have been adding sulphur dioxide to their products. It has antibacterial and antioxidant properties.

Citric Acid:

Citric acid is used to increase the wine's acidity and hence improve the flavour profile or prevent ferric hazes. When a modest amount of citric acid is added to the wine, it enhances the flavour.

Total Sulphur Dioxide:

Total sulphur dioxide levels in a naturally occurring wine are around 10- 20 mg/L.

Density:

Wine is substantially thicker than water in terms of density. Because the amount of residual sugar in sweeter wines is substantially larger than in the rest of the wines, they have a higher density.

Chlorides:

The salt content of the wine is maintained by chlorides. Because of the presence of chloride, the wine has a saline flavour. Though chlorine is a basic element, it is also employed to keep the wine's pH stable.

Fixed acidity:

The fixed acidity of wines includes tartaric, malic, citric and succinic levels. They are used to maintain the sharp flavour of the wine

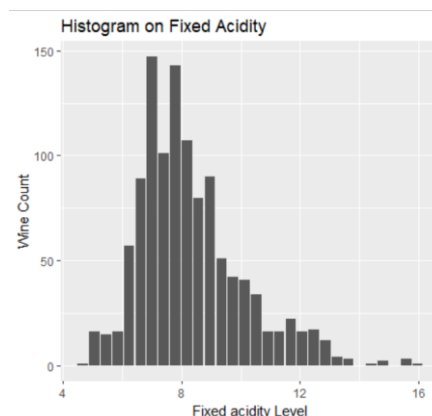


Fig 1: Histogram on Fixed Acidity Level

In the given dataset, the fixed acidity level ranges between 4 to 16. The above histogram can be clustered to form a pattern. Therefore, k means clustering is used.

pH:

All of the wines are on the acidic side of the pH scale. The pH of the majority of the red wines is in the range of 2.5 to 4.5. Furthermore, optimum pH levels must be maintained in order to retain top-quality wine.

Free Sulphur Dioxide:

This parameter aids in the prevention of wine oxidation and microbiological growth. Free sulphur dioxide concentrations of 25 mg/L on red wines throughout maturity and storage.

Residual sugar:

Residual sugar is the amount of sugar left after the fermentation of the wine. Winemakers must create wines with a proper balance of sweetness and sourness. Sweet wines are those that have a residual sugar content of more than 45 grammes per litre.

Quality:

The last but most significant component is the wine's quality. The quality of this dataset goes from 3 to a maximum of 8. Several of the aforementioned characteristics have an impact on the wine's quality.

Data Exploration and Visualisation:

Exploration of data is the first critical step in a data analytics problem. Analysing the data and establishing the relationship between the variables gives a better understanding of what they are dealing with. Several explorations and visualisation of the dataset as been performed and their results are plotted below for a better understanding of the dataset.

Relationship between pH and Alcohol

The acidic or basic quality of a liquid is measured by pH, which stands for "potentially acidic or basic." To keep the wine's quality, it's critical to keep the pH at the proper level. Most wines have a pH of 3 or 4; white wines should have a pH of 3.0 to 3.4, while red wines should have a pH of 3.3 to 3.6.

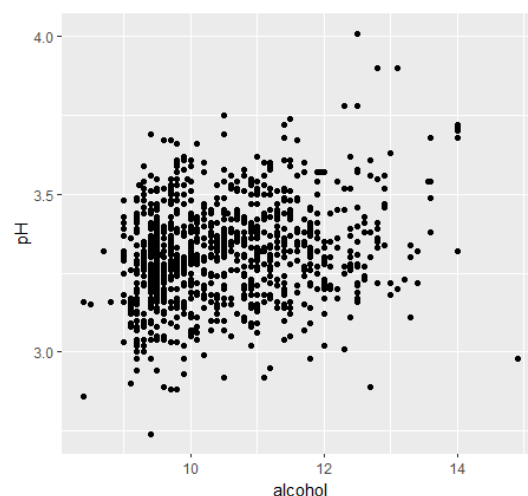


Fig 2: Scatter Plot for Alcohol and pH

Relationship between Quality and alcohol:

A box plot has been built to determine the link between wine quality and alcohol content. The quality of wine improves as the alcohol content rises. The alcohol content of wine with a quality rating of 8 ranges from 9.5 to 14. The alcohol content of grade 3 wines ranges from 8.4 to 11. However, there are few outliers in wine grades 5 and 6.



Fig 3: Box Plot Between Quality and Alcohol Level in Wines

Relationship between Quality and pH:

The accompanying box plot shows the relationship between wine quality and pH range. It's used to visualise the pH ranges in which different types of wine fall. The pH of the highest-quality wines (quality 8) runs from 3 to 3.3, whereas the pH of lower-grade wines goes from 3.1 to 3.55. Furthermore, with the exception of wine quality 3, every other quality has a small number of outliers.

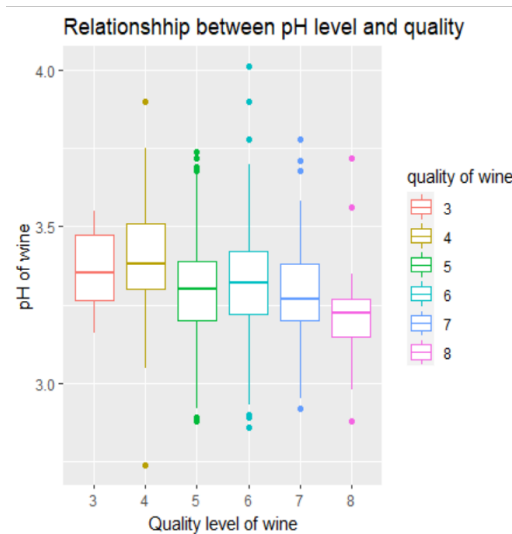


Fig 4: Box Plot Between Quality and pH Level in Wines

Relationship between residual sugar and density of wine:

Wines have a density that is frequently greater than water. This is because the presence of residual sugar in the wine raises the wine's density, which improves its quality. A scatter plot is created to better understand the relationship between density and residual sugar. The graph ranges from 0 to 16. The majority of the points are plotted within the range of 4, but the rest are strewn about the scale in an uneven pattern.

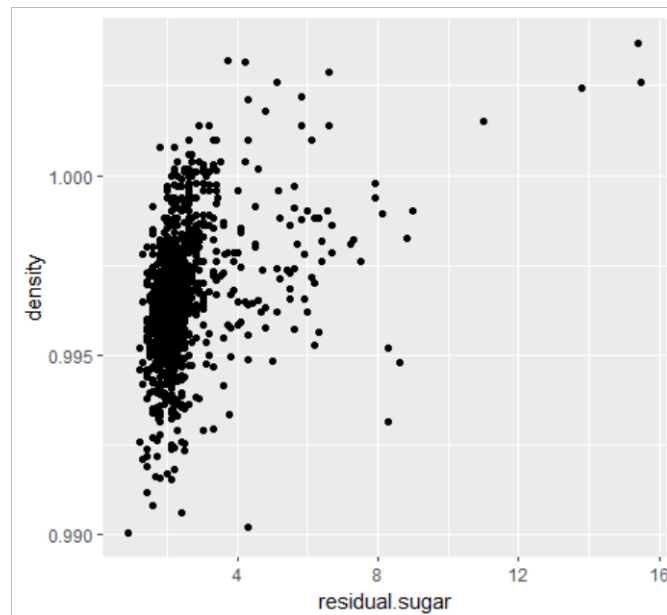


Fig 5: Scatter Plot for Residual Sugar and Density

pH~ Chlorides

Because chloride is a base, it degrades the wine's quality. The bulk of sulphates found in wine are sulphur dioxide molecules and sulphite ions. Many experts believe that wine with a higher sulphur content has a duller flavour and that the high strength of sulphite ions creates a health risk while also speeding up the fermentation process.

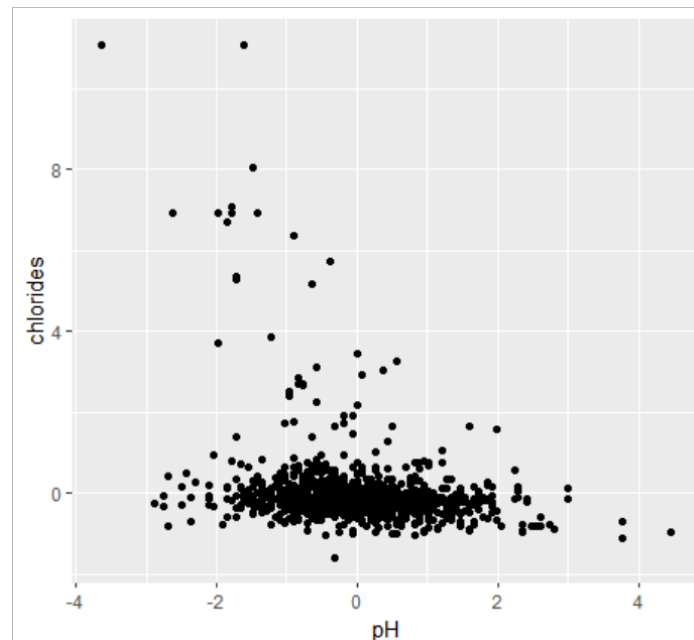


Fig 6: scatter plot for pH and Chlorides

CLUSTERING:

Clustering is a technique for identifying groups of data in a dataset that are similar. The black box effect is used to bring a bunch of data together. Clustering is a type of unsupervised

machine learning. Data is divided into groups using the K means clustering technique based on their distance from the nearest mean or centroid. In addition, the data is divided into k clusters.

K means clustering for fixed acidity:

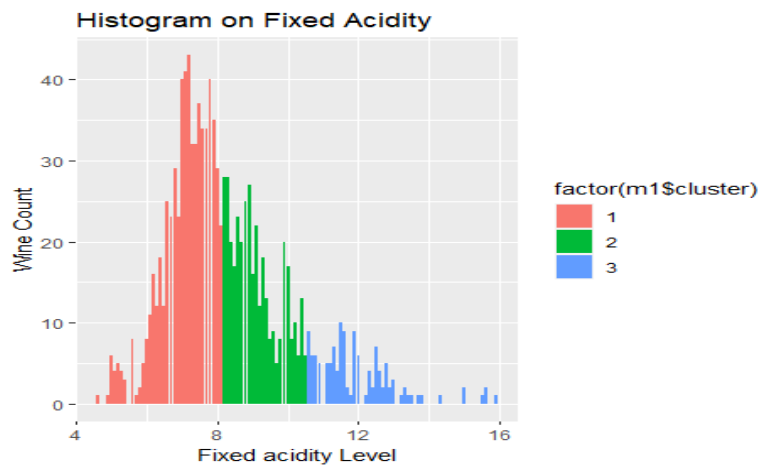


Fig 7: $k = 3$, Clustering on Fixed Acidity of Wines

The k means clustering is used to divide the dataset. The dataset is separated into three clusters since the value of $k = 3$. Cluster 1 represents wines with lower acidity levels, while cluster 2 represents wines with a moderate fixed acidity level and cluster 3 represents wines with a higher fixed acidity level.

K means clustering for residual sugar and density of wine:

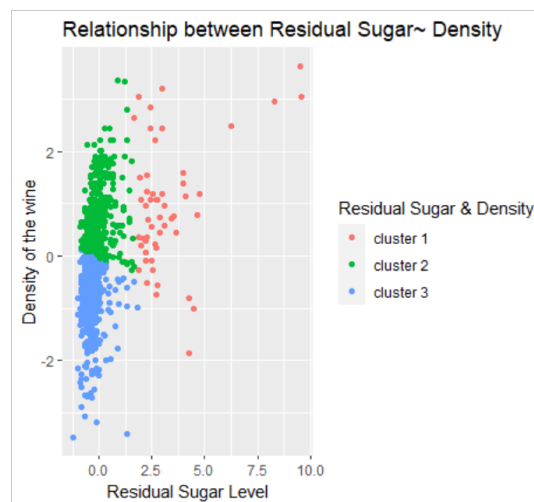


Fig 8: Scatter Plot for Residual Sugar and Density With $k = 3$

With a K value of three, the dataset is initially grouped into three clusters. The data is scaled to increase clustering accuracy. The left-hand massive clump is split into two clusters, while the right-hand outliers are grouped together in a third. As the residual sugar content rises, the wine's density rises as well. The K-means technique is useful for locating the centre points.

K means clustering for pH and Chlorides:

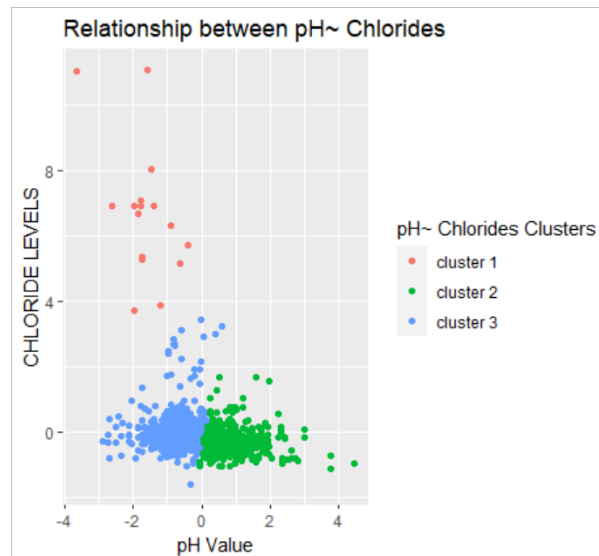


Fig 9: Scatter Plot for pH and chlorides With $k = 3$

The data is scaled, and a scatter plot is constructed to determine the relationship between pH and Chlorides. The k value is set to 3 and the graph is generated as a result. The graph clearly depicts how chloride levels influence the wine's overall pH. When the wine's chlorine level is high, the pH drops; when the chlorine level is low, the pH rises. Because it alters the pH value, it has an impact on the wine's overall quality. Furthermore, it is obvious that the pH of the wine is not only influenced by chlorides, as just a few data points in clusters 2 and 3 have the same chlorine content ratio but are scaled in different pH values.

K means clustering for pH and Alcohol level:

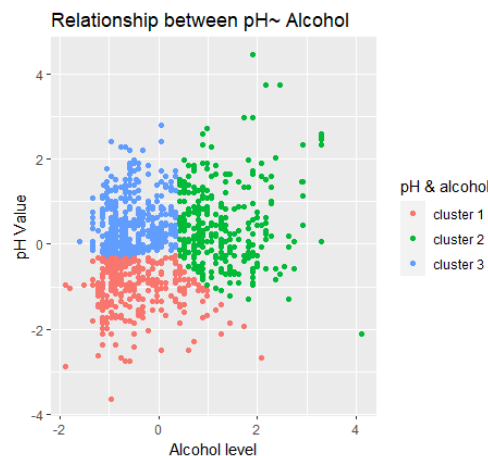
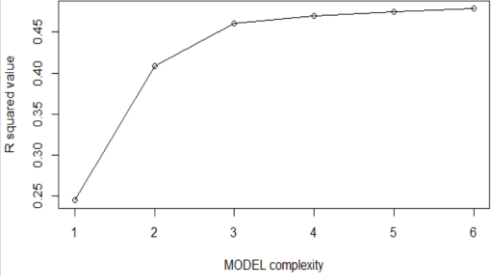
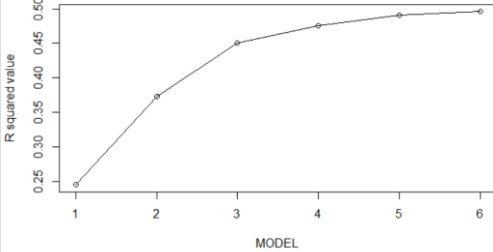


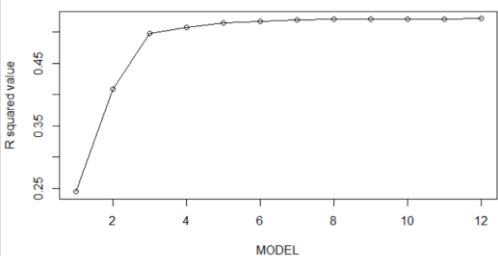
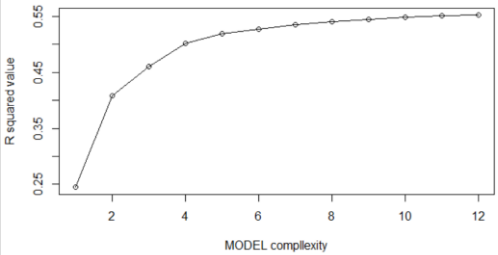
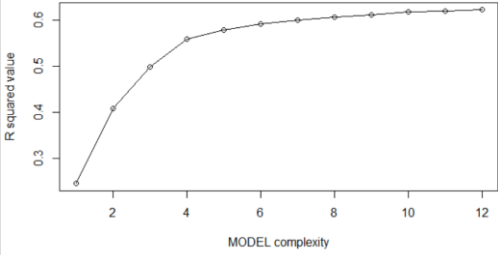
Fig 10: Scatter Plot for pH and alcohol With $k = 3$

Here is a scatter plot for pH and alcohol. To improve clustering results, scaling is used. After scaling, clustering is used to detect patterns. Furthermore, while scaling, the mean and variance are kept at 0 and 1, respectively. As shown in the graph, the data is separated into three groups. In the k means technique, the value of k is kept at 3, and the comparable result is attained. Clusters 1, 2, and 3 each received 450, 375, and 318 points.

REGRESSION MODEL:

The regression model is the first model mentioned, and it is used to build a relationship between multiple variables in the wine dataset. In most cases, the regression model is utilised to forecast the connection between the dependent and independent variables. These variables are referred to be predictors in this context. The link between different predictors is established using multiple predictors. Firstly, before creating a model a subset is to be selected. Generally, there are three different types of predictors, they are namely best subset, forward subset and backward subset. In this case, the best subset is selected in order to provide an optimum relationship between several predictors. Moreover, the best subset selection searches for the most elite model at every iteration. Several regression models were created to establish the relationship between several variables. The output of the regression model is noted in the R-squared value. As the complexity of the model increases, the R-square value also increases. The R square value after which the value tends to stabilise is chosen as the best model.

S.No	Model equation	Corresponding R- square plot	R square value
1	alcohol~	 <p><i>Fig 11: plot between r square value and model complexity of model 1</i></p>	0.46
2	alcohol~ poly(density,3) +poly (residual.sugar, 3)	 <p><i>Fig 12: plot between r square value and model complexity of model 2</i></p>	0.45

3	alcohol~ quality+ poly(pH,4) + poly (volatile.acidity ,3) +chlorides+ poly(density,3)	 <p><i>Fig 13: plot between r square value and model complexity of model 3</i></p>	0.51
4	alcohol~ pH +poly(quality,4) +poly (volatile.acidity ,3) +density + poly(chlorides,3) +poly (residual.sugar, 3)	 <p><i>Fig 14: plot between r square value and model complexity of model 4</i></p>	0.53
5	Alcohol ~ poly (residual.sugar, 3)+poly(quality,3)+poly(density,3) + Poly (volatile.acidity, 2)+poly(pH,3)	 <p><i>Fig 15: plot between r square value and model complexity of model 5</i></p>	0.59

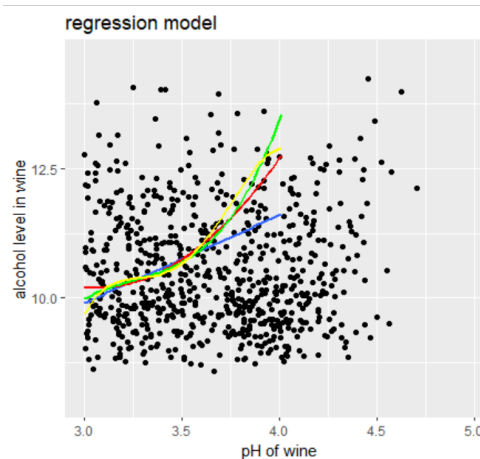


Fig 16: Regression curves between pH and alcohol level in wines

The regression model's graph is constructed using the alcohol content of the wines and their accompanying pH values. The first model is depicted by the blue line, while the second, third, and fourth models are depicted by the red, green, and yellow lines, respectively. The degree of the pH scale has been increased from 1 to 4, and the regression model that corresponds has been plotted. The overfitting of model 4 to the supplied dataset is obvious. From the above-depicted table, model 5 has a higher r square value. Therefore, a minor adjustment to the equation is made to see the variation in results. The predictor's pH and density are iterated to i values for the equation of model 5. The value of i is set to ten in this case. As a result, the equation is transformed into,

$$\text{Alcohol} \sim \text{poly}(\text{residual.sugar}, 3) + \text{poly}(\text{quality}, 3) + \text{poly}(\text{density}, i) + \text{Poly}(\text{volatile.acidity}, 2) + \text{poly}(\text{pH}, i)$$

The corresponding r-square value is identified, and the value is plotted. The chart gives a better R square value than the rest of the models. However, the complexity of the model has substantially increased. However, in this type, all models after model 4 seem to be overfitting the dataset. Therefore, the optimal model for this equation is model 4 with an R square value of 0.63.

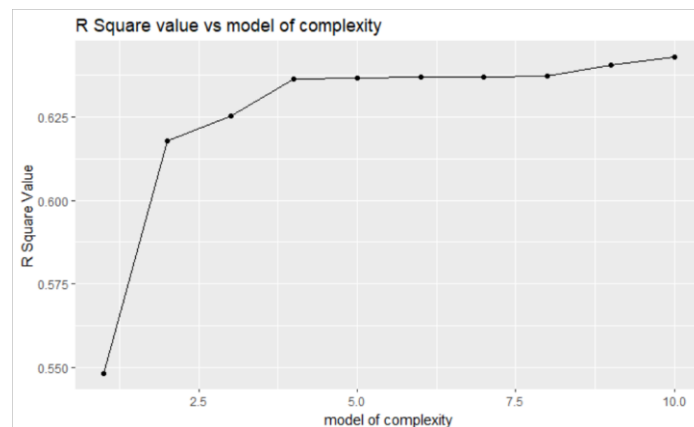


Fig 17: Plot between r square values and model complexity of the altered model 5.

CLASSIFICATION I:

The wine dataset's various predictors can be used to classify wines based on a variety of criteria. In this situation, the wines are categorised according to their quality. It assists wine producers in identifying which wines are the best on the market by rating them according to quality. It enables them to produce higher-quality wines and compete with their competition. Classifying wines based on quality gives a wine connoisseur an advantage in the tasting. It allows them to explore with the wines' flavours. In order to classify wines according to their quality, numerous predictors are used.

Model No.	Classification equation	Success Rate
1	quality~alcohol+pH+residual.sugar+density	58%
2	quality~alcohol+pH+residual.sugar+density+chlorides	58%
3	quality~alcohol+pH+residual.sugar+density+chlorides+volatile.acidity	59%

4	quality~alcohol+pH+residual.sugar+density+chlorides+volatile.acidity+fixed.acidity+total.sulfur.dioxide	60%
5	quality~alcohol+pH+residual.sugar+density+chlorides+volatile.acidity+fixed.acidity+total.sulfur.dioxide+sulphates	61%
6	quality~poly(alcohol,10)+poly(pH,10)+poly(residual.sugar,10)+poly(density,10)+poly(chlorides,10)+poly(volatile.acidity,10)+poly(fixed.acidity,10)+poly(total.sulfur.dioxide,10)+poly(sulphates,10)+poly(free.sulfur.dioxide,10)+poly(citric.acid,10)	59%
7	quality~poly(alcohol,8)+pH+poly(residual.sugar,4)+density+poly(chlorides,6)+poly(volatile.acidity,8)+fixed.acidity+total.sulfur.dioxide+sulphates	58%

In order to classify the wines depending on their quality, multiple predictors are applied. Several models of various complexity were built, and their success rates were calculated. Model 4 has the highest categorization success rate of 60 percent among the models described above. Furthermore, these classification models are subjected to k-fold cross-validation to determine their viability. The value of k is 5 in this case. As a result, the cross-validation model is often known as a 5-fold cross-validation model.

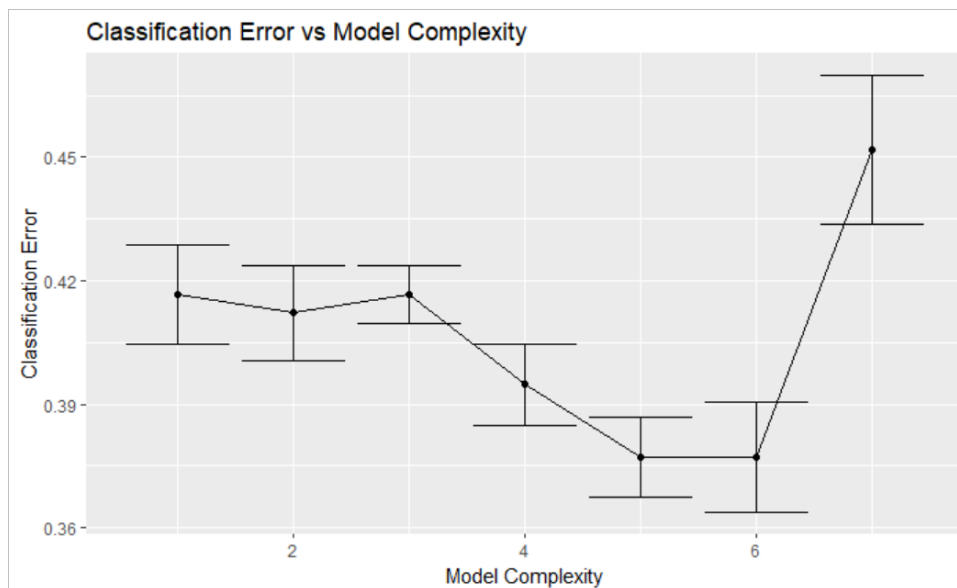


Fig 18: 5-fold error bar graph for classification of wines based on quality

The error range is represented by error bars. Every model has a vertical bar that represents the variation range. The plot clearly shows that model 5 has the smallest standard error. That is, it has the lowest standard error of any other model, at 37 percent. Model 4 is, however, still within the fifth model's variance range. As a result, we can choose model 4 because it has a lower level of complexity than model 5. That is, model 5 has nine predictors, whereas model 4 only has eight. Model 4 is more trustworthy than the other seven types in this case.

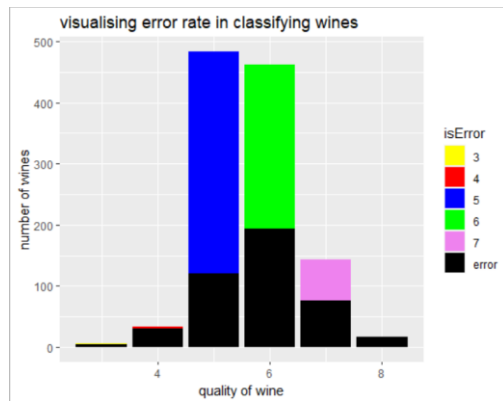


Fig 19: visualising error rates in classifying wines

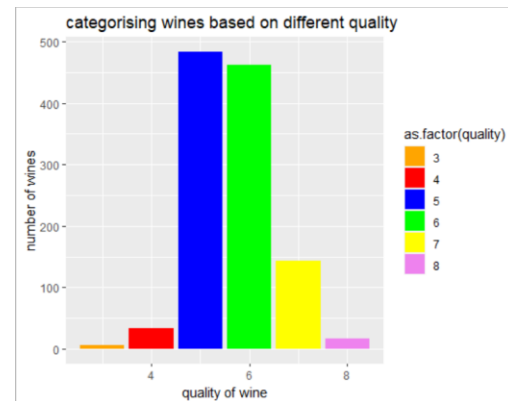


Fig 20: Categorising wines based on different qualities

The above bar chart is used to depict the total number of wines that are falsely classified in separate wine qualities. The falsely classified wines are categorised in black.

The above bar chart depicts the total number of wines classified based on their quality, without considering the error (falsely classified wines).

CLASSIFICATION II:

To improve the classification success rate, a new classification model was developed. [3,4,5,6,7,8,] are the six different categories in which the quality of wine is distributed. As a result, the models struggled to categorise the data into six distinct groups. The quality of the wines is divided into two categories for the convenience of both the wine firms and the consumers. Quality 3,4, and 5 wines are put together in one category, while quality 6,7, and 8 wines are placed together in another. Quality dummy is a new column that categorises them. This prediction serves as a stand-in for the wines' original quality. In the seven-classification model that was created, the quality dummy predictor was used, and the corresponding success rates were obtained.

Model No.	Classification equation	Success rate
1	quality_dummy~alcohol+pH+residual.sugar+density	86%
2	quality_dummy ~alcohol+pH+residual.sugar+density+chlorides	86%
3	quality_dummy~alcohol+pH+residual.sugar+density+chlorides+volatile.acidity	88%
4	quality_dummy~alcohol+pH+residual.sugar+density+chlorides+volatile.acidity+fixed.acidity+total.sulfur.dioxide	86%
5	quality_dummy~alcohol+pH+residual.sugar+density+chlorides+volatile.acidity+fixed.acidity+total.sulfur.dioxide+sulphates	87%
6	quality_dummy~poly(alcohol,10)+poly(pH,10)+poly(residual.sugar,10)+poly(density,10)+poly(chlorides,10)+poly(volatile.acidity,10)+poly(fixed.acidity,10)+poly(total.sulfur.dioxide,10)+poly(sulphates,10)+poly(free.sulfur.dioxide,10)+poly(citric.acid,10)	84%
7	quality_dummy~poly(alcohol,8)+pH+poly(residual.sugar,4)+density+poly(chlorides,6)+poly(volatile.acidity,8)+fixed.acidity+total.sulfur.dioxide+sulphates	84%

The classification success rate has increased significantly as a result of categorising wine quality into two distinct groups. Model 3 has the highest categorization success rate among the models described above, at 88%. Furthermore, these classification models are subjected to k-fold cross-validation to determine their viability. The value of k is 5 in this case. As a result, the cross-validation model is often known as a 5-fold cross-validation model.

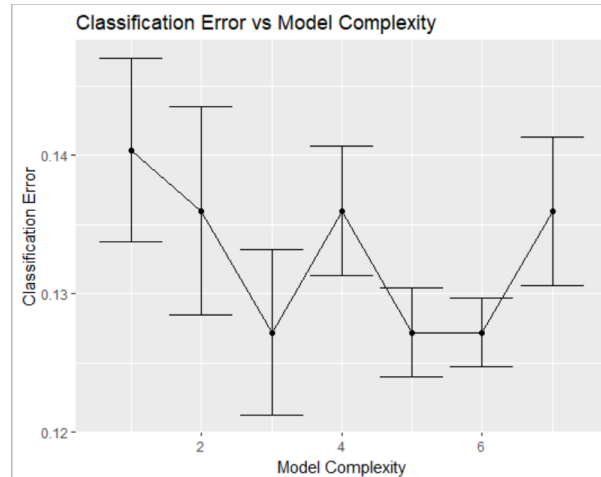


Fig 21: 5-fold error bar graph for classification of wines based on new categorised qualities

The error range is represented by error bars. Every model has a vertical bar that represents the variation range. The plot clearly shows that model 3 has the smallest standard error. That is, it has the lowest standard error of any other model, at 12 percent. Model 2 is, however, still within the third model's variance. As a result, we can choose model 2 because it has a lower level of complexity than model 2. That is, model 3 has six predictors, but model 2 only has five. Model 2 is more trustworthy than the other seven variants in this case.

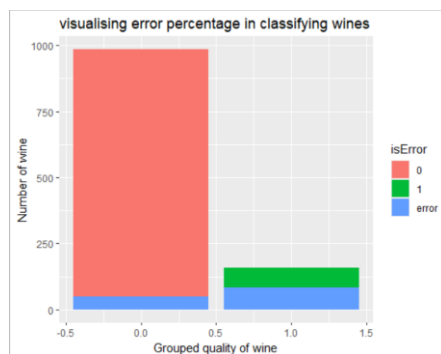


Fig 22: visualising error rates in classifying wines

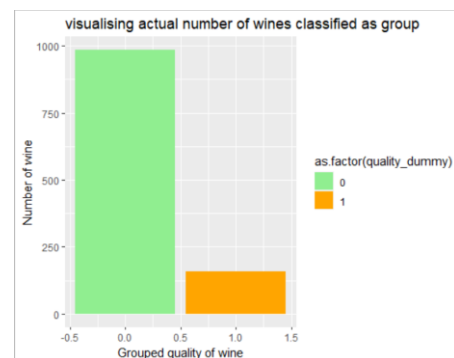


Fig 23: visualising actual number of wines classified as groups

The above bar chart was constructed after sorting the wines into two groups. Wines having a quality rating of 3, 4, and 5 are categorised as 0. Wines having a quality rating of 6, 7, and 8 are categorised as 1. This graph shows the total number of wines that have been classified incorrectly. Wines that have been classed improperly are labelled in blue.

The bar graph above shows the total number of wines classified into two new categories. Wines with a quality grade of 3, 4, and 5 falls under the category of 0. Wines with a quality grade of 6, 7, and 8 falls under the category of 1. It is categorised solely based on its quality, with the inaccuracy factored out (falsely classified wines).

CONCLUSION:

The Portuguese wine dataset was subjected to a data analytics technique. The dataset was first examined to learn the fundamentals of the variables. Furthermore, data visualisation and clustering techniques were used to visualise the association between many variables in the dataset. Multiple regression and classification models were also developed. When compared to the other models in the regression, the fifth model produced the highest r square value. There were two separate classifications carried out. The first was to categorise the wines according to their quality. However, the maximum success rate achieved by the model was 61%. This was achieved by model 5. In practice, this success rate is good because it is higher than the average. However, in order to improve the models' success rate, a new predictor is built in which the quality of wines is divided into two categories: 0 and 1. Wines of quality 3, 4, and 5 are classified as 0; wines of quality 6, 7, and 8 are classified as 1. After then, the classification models are run using the new predictor, and the success rate is recorded. A maximum success rate of 88% has been attained. When categorised quality was employed in classification, Model 3 was the best. Furthermore, the 5-fold cross-validation approach is used to cross-validate these classification models. This allows competitors to see what chemical factors influence the wine's quality. From the consumer's perspective, it allows them to choose from a wide choice of wines based on their quality.

REFERENCE:

1. <https://www.kaggle.com/datasets/yasserh/wine-quality-dataset?datasetId=1866301&searchQuery=clu>
2. <https://www.masterclass.com/articles/what-is-vinho-verde#what-grapes-are-used-to-make-vinho-verde-wines>
3. Professor Scott Flemming's code on clustering. – "Chapter 10 in-class exercise unsupervised learning K-means clustering and Hierarchical clustering" - <https://dal.brightspace.com/d2l/le/content/221958/viewContent/3009644/View>.
4. Professor Scott Flemming code on cross-validation for classification. – "chapter 5 exercise 2 - CV using Classification (LDA and the Iris data set) and the 1SE rule"

APPENDIX A: DATASET:

finalwine.r														
rawwinedata														
	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	alcohol	quality	Id	
1	7.4	0.700	0.00	1.90	0.076	11	34	0.99780	3.51	0.56	9.4	5	0	
2	7.8	0.880	0.00	2.60	0.098	25	67	0.99680	3.20	0.68	9.8	5	1	
3	7.8	0.760	0.04	2.30	0.092	15	54	0.99700	3.26	0.65	9.8	5	2	
4	11.2	0.280	0.56	1.90	0.075	17	60	0.99800	3.16	0.58	9.8	6	3	
5	7.4	0.700	0.00	1.90	0.076	11	34	0.99780	3.51	0.56	9.4	5	4	
6	7.4	0.660	0.00	1.80	0.075	13	40	0.99780	3.51	0.56	9.4	5	5	
7	7.9	0.600	0.06	1.60	0.069	15	59	0.99640	3.30	0.46	9.4	5	6	
8	7.3	0.650	0.00	1.20	0.065	15	21	0.99460	3.39	0.47	10.0	7	7	
9	7.8	0.580	0.02	2.00	0.073	9	18	0.99680	3.36	0.57	9.5	7	8	
10	6.7	0.580	0.08	1.80	0.097	15	65	0.99590	3.28	0.54	9.2	5	10	
11	5.6	0.615	0.00	1.60	0.089	16	59	0.99430	3.58	0.52	9.9	5	12	
12	7.8	0.610	0.29	1.60	0.114	9	29	0.99740	3.26	1.56	9.1	5	13	
13	8.5	0.280	0.56	1.80	0.092	35	103	0.99690	3.30	0.75	10.5	7	16	
14	7.9	0.320	0.51	1.80	0.341	17	56	0.99690	3.04	1.08	9.2	6	19	
15	7.6	0.390	0.31	2.30	0.082	23	71	0.99820	3.52	0.65	9.7	5	21	
16	7.9	0.430	0.21	1.60	0.106	10	37	0.99660	3.17	0.91	9.5	5	22	
17	8.5	0.490	0.11	2.30	0.084	9	67	0.99680	3.17	0.53	9.4	5	23	
18	6.9	0.400	0.14	2.40	0.085	21	40	0.99680	3.43	0.63	9.7	6	24	
19	6.3	0.390	0.16	1.40	0.080	11	23	0.99550	3.34	0.56	9.3	5	25	

Showing 1 to 19 of 1,143 entries, 13 total columns

Fig 24: Screenshot of the wine dataset with the actual quality

	fixed.acidity	volatile.acidity	citric.acid	residual.sugar	chlorides	free.sulfur.dioxide	total.sulfur.dioxide	density	pH	sulphates	alcohol	quality	Id	quality_dummy
1	7.4	0.700	0.00	1.90	0.076	11	34	0.99780	3.51	0.56	9.4	5	0	0
2	7.8	0.880	0.00	2.60	0.098	25	67	0.99680	3.20	0.68	9.8	5	1	0
3	7.8	0.760	0.04	2.30	0.092	15	54	0.99700	3.26	0.65	9.8	5	2	0
4	11.2	0.280	0.56	1.90	0.075	17	60	0.99800	3.16	0.58	9.8	6	3	0
5	7.4	0.700	0.00	1.90	0.076	11	34	0.99780	3.51	0.56	9.4	5	4	0
6	7.4	0.660	0.00	1.80	0.075	13	40	0.99780	3.51	0.56	9.4	5	5	0
7	7.9	0.600	0.06	1.60	0.069	15	59	0.99640	3.30	0.46	9.4	5	6	0
8	7.3	0.650	0.00	1.20	0.065	15	21	0.99460	3.39	0.47	10.0	7	7	1
9	7.8	0.580	0.02	2.00	0.073	9	18	0.99680	3.36	0.57	9.5	7	8	1
10	6.7	0.580	0.08	1.80	0.097	15	65	0.99590	3.28	0.54	9.2	5	10	0
11	5.6	0.615	0.00	1.60	0.089	16	59	0.99430	3.58	0.52	9.9	5	12	0
12	7.8	0.610	0.29	1.60	0.114	9	29	0.99740	3.26	1.56	9.1	5	13	0
13	8.5	0.280	0.56	1.80	0.092	35	103	0.99690	3.30	0.75	10.5	7	16	1
14	7.9	0.320	0.51	1.80	0.341	17	56	0.99690	3.04	1.08	9.2	6	19	0
15	7.6	0.390	0.31	2.30	0.082	23	71	0.99820	3.52	0.65	9.7	5	21	0
16	7.9	0.430	0.21	1.60	0.106	10	37	0.99660	3.17	0.91	9.5	5	22	0
17	8.5	0.490	0.11	2.30	0.084	9	67	0.99680	3.17	0.53	9.4	5	23	0
18	6.9	0.400	0.14	2.40	0.085	21	40	0.99680	3.43	0.63	9.7	6	24	0

Fig 25: Screenshot of the wine dataset with categorised quality

APPENDIX B: PRIOR PROBABILITIES:

CLASSIFICATION I: Prior probability of the seven models before categorising the qualities into 0 and 1.

Model 1: `lda(quality~alcohol+pH+residual.sugar+density)`

Prior probabilities of groups:

3 4 5 6 7 8
0.006993007 0.031468531 0.393356643 0.423076923 0.131118881 0.013986014

Model 2: `lda(quality~alcohol+pH+residual.sugar+density+chlorides)`

Prior probabilities of groups:

3 4 5 6 7 8
0.006993007 0.031468531 0.393356643 0.423076923 0.131118881 0.013986014

Model 3: `lda(quality~alcohol+pH+residual.sugar+density+chlorides+volatile.acidity)`

Prior probabilities of groups:

3 4 5 6 7 8
0.006993007 0.031468531 0.393356643 0.423076923 0.131118881 0.013986014

Model 4:

`lda(quality~alcohol+pH+residual.sugar+density+chlorides+volatile.acidity+fixed.acidity+total.sulfur.dioxide)`

Prior probabilities of groups:

3 4 5 6 7 8
0.006993007 0.031468531 0.393356643 0.423076923 0.131118881 0.013986014

Model 5:

`lda(quality~alcohol+pH+residual.sugar+density+chlorides+volatile.acidity+fixed.acidity+total.sulfur.dioxide+sulphates)`

Prior probabilities of groups:

3	4	5	6	7	8
0.006993007	0.031468531	0.393356643	0.423076923	0.131118881	0.013986014

Model 6:

lda(quality~poly(alcohol,10)+poly(pH,10)+poly(residual.sugar,10)+poly(density,10)+poly(chlorides,10)+poly(volatile.acidity,10)+poly(fixed.acidity,10)+poly(total.sulfur.dioxide,10)+poly(sulphates,10)+poly(free.sulfur.dioxide,10)+poly(citric.acid,10))

Prior probabilities of groups:

3	4	5	6	7	8
0.006993007	0.031468531	0.393356643	0.423076923	0.131118881	0.013986014

Model 7:

lda(quality~poly(alcohol,8)+pH+poly(residual.sugar,4)+density+poly(chlorides,6)+poly(volatile.acidity,8)+fixed.acidity+total.sulfur.dioxide+sulphates)

Prior probabilities of groups:

3	4	5	6	7	8
0.006993007	0.031468531	0.393356643	0.423076923	0.131118881	0.013986014

CLASSIFICATION II:

Prior probability of the seven models after categorising the qualities into 0 and 1.

Model 1: lda(quality~alcohol+pH+residual.sugar+density)

Prior probabilities of groups:

0	1
0.8548951	0.1451049

Model 2: lda(quality~alcohol+pH+residual.sugar+density+chlorides)

Prior probabilities of groups:

0	1
0.8548951	0.1451049

Model 3: lda(quality~alcohol+pH+residual.sugar+density+chlorides+volatile.acidity)

Prior probabilities of groups:

0	1
0.8548951	0.1451049

Model 4:

lda(quality~alcohol+pH+residual.sugar+density+chlorides+volatile.acidity+fixed.acidity+total.sulfur.dioxide)

Prior probabilities of groups:

0	1
0.8548951	0.1451049

Model 5:

```
lda(quality~alcohol+pH+residual.sugar+density+chlorides+volatile.acidity+fixed.acidity+total.sulfur.dioxide+sulphates
```

Prior probabilities of groups:

```
      0      1  
0.8548951 0.1451049
```

Model 6:

```
lda(quality~poly(alcohol,10)+poly(pH,10)+poly(residual.sugar,10)+poly(density,10)+poly(chlorides,10)+poly(volatile.acidity,10)+poly(fixed.acidity,10)+poly(total.sulfur.dioxide,10)+poly(sulphates,10)+poly(free.sulfur.dioxide,10)+poly(citric.acid,10)
```

Prior probabilities of groups:

```
      0      1  
0.8548951 0.1451049
```

Model 7:

```
lda(quality~poly(alcohol,8)+pH+poly(residual.sugar,4)+density+poly(chlorides,6)+poly(volatile.acidity,8)+fixed.acidity+total.sulfur.dioxide+sulphates)
```

Prior probabilities of groups:

```
      0      1  
0.8548951 0.1451049
```

R CODE:

```
library(tidyverse)
```

```
library(dplyr)
```

```
library(ggplot2)
```

```
library(stats)
```

```
library(scales)
```

```
library(leaps)
```

```
library(MASS)
```

```
#importing dataset
```

```
setwd("C:/CANADA/industrial engineering/summer term/data analytics/dataset")
```

```
rawwinedata= read.csv("wineQT.csv")
```

```
head(rawwinedata)
```

```
view(rawwinedata)
```

```
str(rawwinedata)
```

```

dim(rawwinedata)

#cleaning dataset
colSums(is.na(rawwinedata))
cleanwinedata <- rawwinedata[complete.cases(rawwinedata),]
dim(cleanwinedata)
view(cleanwinedata)

#regression model:

attach(cleanwinedata)

wine= dplyr::select(cleanwinedata, alcohol,quality, residual.sugar,density, volatile.acidity,
pH, chlorides)
#model 1
best = regsubsets(alcohol~., data=wine)
summary(best)
summary(best)$rsq
plot(summary(best)$rsq, type = "o",xlab = "MODEL complexity",ylab = "R squared value")
#0.4604

#model2
best2 = regsubsets(alcohol~poly(density,3)+poly(residual.sugar,3),data= wine,nvmax = 12)
summary(best2)
summary(best2)$rsq
plot(summary(best2)$rsq, type = "o",xlab = "MODEL",ylab = "R squared value")
#0.45

#model3
best3 = regsubsets(alcohol~quality+poly(pH,4)+poly(volatile.acidity,3)
+chlorides+poly(density,3),data= wine,nvmax = 12)

```

```

summary(best3)

summary(best3)$rsq

plot(summary(best3)$rsq, type = "o", xlab = "MODEL", ylab = "R squared value")

#0.51


#model4

best4 = regsubsets(alcohol~pH+poly(quality,4)+poly(volatile.acidity,3)
                  +density+poly(chlorides,3)+poly(residual.sugar,3), data= wine, nvmax = 12)

summary(best4)

summary(best4)$rsq

plot(summary(best4)$rsq, type = "o", xlab = "MODEL complexity", ylab = "R squared
value")

#0.5348


#model5

wbest5 = regsubsets(alcohol~poly(residual.sugar,3)+poly(quality,3)+poly(density,3)+
                  poly(volatile.acidity,2)+poly(pH,3),
                  data= wine, nvmax = 12)

summary(wbest5)

summary(wbest5)$rsq

plot(summary(wbest5)$rsq, type = "o", xlab = "MODEL complexity", ylab = "R squared
value")

#0.59


ggplot(data=wine, aes(y=alcohol,x=pH))+
  geom_point(position = position_jitter(width = 1, height = .5))+
  geom_smooth(method="lm", se = FALSE)+
  geom_smooth(method="lm", formula= y~poly(x,2), colour= "red", se = FALSE)+
  geom_smooth(method="lm", formula= y~poly(x,3), colour= "green", se = FALSE)+
  geom_smooth(method="lm", formula= y~poly(x,4), colour= "yellow", se = FALSE)+
  xlab("pH of wine")+

```

```
ylab("alcohol level in wine")+
ggtitle("regression model")+
xlim(3, 5)
```

```
rval= seq(1,10)
rvalx= seq(1,10)
for (i in 1:10) {
  model= lm(data=wine,alcohol~poly(residual.sugar,3)+poly(quality,3)+poly(density,i)+
    poly(volatile.acidity,2)+poly(pH,i))
  rval[i]= summary(model)$r.squared
}
view(rval)
```

```
modelR2val= data.frame(rvalx,rval)
ggplot(data=modelR2val)+
  geom_point(aes(x=rvalx,y=rval))+
  geom_path(x=rvalx,y=rval)+
  xlab("model of complexity")+
  ylab("R Square Value")+
  ggtitle("R Square value vs model of complexity")
```

```
#group plot
library(GGally)
attach(cleanwinedata)
cleanwinedata$quality <- as.factor(cleanwinedata$quality)
ggpairs(cleanwinedata, aes(colour = quality))
```

```
#linear discriminant analysis
```

```

attach(cleanwinedata)

library(MASS)

#splitting training and test data


view(cleanwinedata)

IDwine = mutate(cleanwinedata, id=row_number())
View(IDwine)

winetrainDataSet = sample_frac(IDwine, .5)
View(winetrainDataSet)

winetestDataSet = anti_join(IDwine, winetrainDataSet, by = "id")
View(winetestDataSet)


#creating first model
view(winetestDataSet)

winemodel1 = lda(quality~alcohol+pH+residual.sugar+density, data = winetrainDataSet)
winemodel1
winetrainSetPrediction1 = predict(winemodel1, winetrainDataSet)
table(winetrainSetPrediction1$class, winetrainDataSet$quality)
mean(winetrainSetPrediction1$class== winetrainDataSet$quality)


#running model for test data
wineTestSetprediction1 = predict(winemodel1,winetestDataSet )
table(wineTestSetprediction1$class , winetestDataSet$quality)
mean(wineTestSetprediction1$class == winetestDataSet$quality)


#creating second model
winemodel2 = lda(quality~alcohol+pH+residual.sugar+density+chlorides, data =
winetrainDataSet)
winemodel2
winetrainSetPrediction2 = predict(winemodel2, winetrainDataSet)
table(winetrainSetPrediction2$class, winetrainDataSet$quality)

```



```
mean(winetrainSetPrediction2$class== winetrainDataSet$quality)
```

```
#running model for test data
```

```
wineTestSetprediction2 = predict(winemodel2,winetestDataSet )
```

```
table(wineTestSetprediction2$class , winetestDataSet$quality)
```

```
mean(wineTestSetprediction2$class == winetestDataSet$quality)
```

```
#creating third model
```

```
winemodel3 = lda(quality~alcohol+pH+residual.sugar+density+chlorides+  
                volatile.acidity, data = winetrainDataSet)
```

```
winemodel3
```

```
winetrainSetPrediction3 = predict(winemodel3, winetrainDataSet)
```

```
table(winetrainSetPrediction3$class, winetrainDataSet$quality)
```

```
mean(winetrainSetPrediction3$class== winetrainDataSet$quality)
```

```
#running model for test data
```

```
wineTestSetprediction3 = predict(winemodel3,winetestDataSet )
```

```
table(wineTestSetprediction3$class , winetestDataSet$quality)
```

```
mean(wineTestSetprediction3$class == winetestDataSet$quality)
```

```
#0.57
```

```
#creating fourth model
```

```
winemodel4 = lda(quality~alcohol+pH+residual.sugar+density+chlorides+  
                volatile.acidity+fixed.acidity+total.sulfur.dioxide  
                , data = winetrainDataSet)
```

```
winemodel4
```

```
winetrainSetPrediction4 = predict(winemodel4, winetrainDataSet)
```

```
table(winetrainSetPrediction4$class, winetrainDataSet$quality)
```

```
mean(winetrainSetPrediction4$class== winetrainDataSet$quality)
```

```
#running model for test data
wineTestSetprediction4 = predict(winemodel4,winetestDataSet )
table(wineTestSetprediction4$class , winetestDataSet$quality)
mean(wineTestSetprediction4$class == winetestDataSet$quality)
#0.60
```

```
#creating fifth model
winemodel5 = lda(quality~alcohol+pH+residual.sugar+density+chlorides+
                 volatile.acidity+fixed.acidity+total.sulfur.dioxide
                 +sulphates, data = winetrainDataSet)
winemodel5
winetrainSetPrediction5 = predict(winemodel5, winetrainDataSet)
table(winetrainSetPrediction5$class, winetrainDataSet$quality)
mean(winetrainSetPrediction5$class== winetrainDataSet$quality)
```

```
#running model for test data
wineTestSetprediction5 = predict(winemodel5,winetestDataSet )
table(wineTestSetprediction5$class , winetestDataSet$quality)
mean(wineTestSetprediction5$class == winetestDataSet$quality)
#0.58
```

```
#creating sixth model
winemodel6 = lda(quality~poly(alcohol,10)+poly(pH,10)+poly(residual.sugar,10)
                 +poly(density,10)+poly(chlorides,10)+poly(volatile.acidity,10)
                 +poly(fixed.acidity,10)+poly(total.sulfur.dioxide,10)
                 +poly(sulphates,10)+ poly(free.sulfur.dioxide,10)+poly(citric.acid,10), data =
winetrainDataSet)
winemodel6
winetrainSetPrediction6 = predict(winemodel6, winetrainDataSet)
table(winetrainSetPrediction6$class, winetrainDataSet$quality)
mean(winetrainSetPrediction6$class== winetrainDataSet$quality)
```

```

#running model for test data
wineTestSetprediction6 = predict(winemodel6,winetestDataSet )
table(wineTestSetprediction6$class , winetestDataSet$quality)
mean(wineTestSetprediction6$class == winetestDataSet$quality)
#0.56

#creating seventh model
winemodel7 = lda(quality~poly(alcohol,8)+pH+poly(residual.sugar,4)
  +density+poly(chlorides,6)
  +poly(volatile.acidity,8)+fixed.acidity+total.sulfur.dioxide
  +sulphates, data = winetrainDataSet)
winemodel7
winetrainSetPrediction7 = predict(winemodel7, winetrainDataSet)
table(winetrainSetPrediction7$class, winetrainDataSet$quality)
mean(winetrainSetPrediction7$class== winetrainDataSet$quality)
#running model for test data
wineTestSetprediction7 = predict(winemodel7,winetestDataSet )
table(wineTestSetprediction7$class , winetestDataSet$quality)
mean(wineTestSetprediction7$class == winetestDataSet$quality)

#creating conclusion graph to visualize the error.
attach(cleanwinedata)
model1D = lda(quality~poly(alcohol)+pH+residual.sugar+density+chlorides+
  volatile.acidity+fixed.acidity+total.sulfur.dioxide
  +sulphates+ free.sulfur.dioxide+citric.acid)
guess1D = predict(model1D, cleanwinedata)
winewith1Dguesses = mutate(cleanwinedata, guess=guess1D$class,
  isError = ifelse(guess==quality, as.character(guess),
    "error"))
View(winewith1Dguesses)

```

```
ggplot(data= winewith1Dguesses )+geom_bar( aes(x= quality,
fill=isError))+scale_fill_manual(values = c("yellow", "red",
"blue", "green", "violet", "black"))+

ylab("number of wines")+
xlab("quality of wine")+
labs(color = "outcomes")+
ggtitle("visualising error rate in classifying wines")
```

```
ggplot(data=winewith1Dguesses)+geom_bar(aes(x= quality,
fill=as.factor(quality)))+scale_fill_manual(values = c( "orange", "red",
"blue", "green", "yellow", "violet"))+
ylab("number of wines")+
xlab("quality of wine")+
labs(color = "outcomes")+
ggtitle("categorising wines based on different quality")
```

```
#classification cross validation
```

```
wineMix = slice(cleanwinedata, sample(1:n()))
```

```
summary(cleanwinedata)
```

```
summary(wineMix)
```

```
id = seq(1, 1143, by=1)
```

```
wineRando = mutate(wineMix, id)
```

```
k = 5
```

```
numRows = nrow(cleanwinedata)
```

```
train = filter(wineRando, id <= 4*numRows/k)
```

```
test = anti_join(wineRando, train, by="id")
```

```
View(train)
```

```
View(test)
```

```
library(MASS)
```

```
attach(cleanwinedata)
```

```
k = 5
```

```
numRows = nrow(cleanwinedata)
```

```
errors_1 = rep(0, k)
```

```
totalError = 0
```

```
for(i in 1:k){
```

```
  test = filter(wineRando, id >= (i-1)*numRows/k+1 & id <= i*numRows/k)
```

```

train = anti_join(wineRando, test, by="id")

model = lda(quality~alcohol+pH+residual.sugar+density, train)

modelGuesses = predict(model, test)

errors_1[i] = 1-mean(modelGuesses$class == test$quality)

totalError = errors_1[i]+totalError

}

errors_1

totalError

avgerror=totalError/k

avgerror

#second model
errors_2 = rep(0, k)

totalError = 0

for(i in 1:k){

test = filter(wineRando, id >= (i-1)*numRows/k+1 & id <= i*numRows/k)

```

```
train = anti_join(wineRando, test, by="id")
```

```
model = lda(quality~alcohol+pH+residual.sugar+density+chlorides, train)
```

```
modelGuesses = predict(model, test)
```

```
errors_2[i] = 1-mean(modelGuesses$class == test$quality)
```

```
totalError = errors_2[i]+totalError
```

```
}
```

```
errors_2
```

```
totalError
```

```
avgerror=totalError/k
```

```
avgerror
```

```
#third model
```

```
errors_3 = rep(0, k)
```

```
totalError = 0
```

```
for(i in 1:k){
```

```

test = filter(wineRando, id >= (i-1)*numRows/k+1 & id <= i*numRows/k)

train = anti_join(wineRando, test, by="id")

model = lda(quality~alcohol+pH+residual.sugar+density+chlorides+
            volatile.acidity, train)

modelGuesses = predict(model, test)

errors_3[i] = 1-mean(modelGuesses$class == test$quality)

totalError = errors_3[i]+totalError

}

errors_3

totalError

avgerror=totalError/k

avgerror

#fourth model

errors_4 = rep(0, k)

totalError = 0

for(i in 1:k){

```



```
test = filter(wineRando, id >= (i-1)*numRows/k+1 & id <= i*numRows/k)
```

```
train = anti_join(wineRando, test, by="id")
```

```
model = lda(quality~alcohol+pH+residual.sugar+density+chlorides+  
            volatile.acidity+fixed.acidity+total.sulfur.dioxide, train)
```

```
modelGuesses = predict(model, test)
```

```
errors_4[i] = 1-mean(modelGuesses$class == test$quality)
```

```
totalError = errors_4[i]+totalError
```

```
}
```

```
errors_4
```

```
totalError
```

```
avgerror=totalError/k
```

```
avgerror
```

```
#fifth model
```

```
errors_5 = rep(0, k)
```

```
totalError = 0
```

```

for(i in 1:k){

  test = filter(wineRando, id >= (i-1)*numRows/k+1 & id <= i*numRows/k)

  train = anti_join(wineRando, test, by="id")

  model = lda(quality~alcohol+pH+residual.sugar+density+chlorides+
              volatile.acidity+fixed.acidity+total.sulfur.dioxide
              +sulphates, train)

  modelGuesses = predict(model, test)

  errors_5[i] = 1-mean(modelGuesses$class == test$quality)

  totalError = errors_5[i]+totalError

}

errors_5

totalError

avgerror=totalError/k

avgerror

#sixth model

errors_6 = rep(0, k)

```

```

totalError = 0

for(i in 1:k){

  test = filter(wineRando, id >= (i-1)*numRows/k+1 & id <= i*numRows/k)

  train = anti_join(wineRando, test, by="id")

  model =
lda(quality~poly(alcohol,10)+poly(pH,10)+poly(residual.sugar,10)+poly(density,10)+poly(ch
lorides,10)+poly(volatile.acidity,10)
      +poly(fixed.acidity,10)+poly(total.sulfur.dioxide,10)+poly(sulphates,10)+
poly(free.sulfur.dioxide,10)+poly(citric.acid,10)
      , train)

  modelGuesses = predict(model, test)

  errors_6[i] = 1-mean(modelGuesses$class == test$quality)

  totalError = errors_6[i]+totalError

}

errors_6

totalError

avgerror=totalError/k

avgerror

```

```

#seventh model

errors_7 = rep(0, k)

totalError = 0

for(i in 1:k){

  test = filter(wineRando, id >= (i-1)*numRows/k+1 & id <= i*numRows/k)

  train = anti_join(wineRando, test, by="id")

  model =
lda(quality~poly(alc0hol,8)+pH+poly(residual.sugar,4)+density+poly(chlorides,6)+poly(vola
tile.acidity,8)+fixed.acidity+total.sulfur.dioxide
      +sulphates
      , train)

  modelGuesses = predict(model, test)

  errors_7[i] = 1-mean(modelGuesses$class == test$quality)

  totalError = errors_7[i]+totalError

}

errors_7

totalError

```

```
avgerror=totalError/k
```

```
avgerror
```

```
errors_1
```

```
errors_2
```

```
errors_3
```

```
errors_4
```

```
errors_5
```

```
errors_6
```

```
errors_7
```

```
avgE=rep(0,7)
```

```
avgE
```

```
for(i in 1:k){
```

```
  avgE[1]=errors1[i]+avgE[1]
```

```
  avgE[2]=errors2[i]+avgE[2]
```

```
  avgE[3]=errors3[i]+avgE[3]
```

avgE[4]=errors4[i]+avgE[4]

avgE[5]=errors5[i]+avgE[5]

avgE[6]=errors6[i]+avgE[6]

avgE[7]=errors7[i]+avgE[7]

}

avgE[1]

avgE[2]

avgE[3]

avgE[4]

avgE[5]

avgE[6]

avgE[7]

se=rep(0,k)

se[1]=sqrt(var(errors1)/k)

```
se[2]=sqrt(var(errors2)/k)
```

```
se[3]=sqrt(var(errors3)/k)
```

```
se[4]=sqrt(var(errors4)/k)
```

```
se[5]=sqrt(var(errors5)/k)
```

```
se[6]=sqrt(var(errors6)/k)
```

```
se[7]=sqrt(var(errors7)/k)
```

```
mn=seq(1,7, by=1)
```

```
length(avgE)
```

```
length(se)
```

```
cross_validation = data.frame(avgE, se)
```

```
View(cross_validation)
```

```
#plotting data
```

```
library(tidyverse)
```

```
ggplot(cross_validation, aes(x=mn,y=avgE))+
```

```
  geom_line()+
```

```

geom_point()+

geom_errorbar(aes(ymin=avgE-se, ymax=avgE+se))+

xlab("Model Complexity")+

ylab("Classification Error")+

ggtitle("Classification Error vs Model Complexity ")

#Classification II

cleanwinedata$quality_dummy <- ifelse(wine$quality > 6, 1, 0)

view(cleanwinedata)

idwine = mutate(cleanwinedata, id=row_number())

View(idwine)

winetrainDataSet = sample_frac(idwine, .5)

View(winetrainDataSet)

winetestDataSet = anti_join(idwine, winetrainDataSet, by = "id")

View(winetestDataSet)

#creating first model

view(winetestDataSet)

winemodel1 = lda(quality_dummy~alcohol+pH+residual.sugar+density, data =
winetrainDataSet)

winemodel1

winetrainSetPrediction1 = predict(winemodel1, winetrainDataSet)

table(winetrainSetPrediction1$class, winetrainDataSet$quality_dummy)

mean(winetrainSetPrediction1$class== winetrainDataSet$quality_dummy)

```



```
#running model for test data
```

```
wineTestSetprediction1 = predict(winemodel1,winetestDataSet )
```

```
table(wineTestSetprediction1$class , winetestDataSet$quality_dummy)
```

```
mean(wineTestSetprediction1$class == winetestDataSet$quality_dummy)
```

```
#0.86
```

```
#creating second model
```

```
winemodel2 = lda(quality_dummy~alcohol+pH+residual.sugar+density+chlorides, data =  
winetrainDataSet)
```

```
winemodel2
```

```
winetrainSetPrediction2 = predict(winemodel2, winetrainDataSet)
```

```
table(winetrainSetPrediction2$class, winetrainDataSet$quality_dummy)
```

```
mean(winetrainSetPrediction2$class== winetrainDataSet$quality_dummy)
```

```
#running model for test data
```

```
wineTestSetprediction2 = predict(winemodel2,winetestDataSet )
```

```
table(wineTestSetprediction2$class , winetestDataSet$quality_dummy)
```

```
mean(wineTestSetprediction2$class == winetestDataSet$quality_dummy)
```

```
#0.86
```

```
#creating third model
```

```
winemodel3 = lda(quality_dummy~alcohol+pH+residual.sugar+density+chlorides+  
volatile.acidity, data = winetrainDataSet)
```

```
winemodel3
```

```
winetrainSetPrediction3 = predict(winemodel3, winetrainDataSet)
```

```
table(winetrainSetPrediction3$class, winetrainDataSet$quality_dummy)
```

```
mean(winetrainSetPrediction3$class== winetrainDataSet$quality_dummy)
```

```
#running model for test data
```

```
wineTestSetprediction3 = predict(winemodel3,winetestDataSet )
```

```

table(wineTestSetprediction3$class , winetestDataSet$quality_dummy)
mean(wineTestSetprediction3$class == winetestDataSet$quality_dummy)
#0.88

#creating fourth model
winemodel4 = lda(quality_dummy~alcohol+pH+residual.sugar+density+chlorides+
                volatile.acidity+fixed.acidity+total.sulfur.dioxide
                , data = winetrainDataSet)
winemodel4
winetrainSetPrediction4 = predict(winemodel4, winetrainDataSet)
table(winetrainSetPrediction4$class, winetrainDataSet$quality_dummy)
mean(winetrainSetPrediction4$class== winetrainDataSet$quality_dummy)

#running model for test data
wineTestSetprediction4 = predict(winemodel4,winetestDataSet )
table(wineTestSetprediction4$class , winetestDataSet$quality_dummy)
mean(wineTestSetprediction4$class == winetestDataSet$quality_dummy)
#0.86

#creating fifth model
winemodel5 = lda(quality_dummy~alcohol+pH+residual.sugar+density+chlorides+
                volatile.acidity+fixed.acidity+total.sulfur.dioxide
                +sulphates, data = winetrainDataSet)
winemodel5
winetrainSetPrediction5 = predict(winemodel5, winetrainDataSet)
table(winetrainSetPrediction5$class, winetrainDataSet$quality_dummy)
mean(winetrainSetPrediction5$class== winetrainDataSet$quality_dummy)

#running model for test data
wineTestSetprediction5 = predict(winemodel5,winetestDataSet )

```

```

table(wineTestSetprediction5$class , winetestDataSet$quality_dummy)
mean(wineTestSetprediction5$class == winetestDataSet$quality_dummy)
#0.87

#creating sixth model
winemodel6 = lda(quality_dummy~poly(alcohol,10)+poly(pH,10)+poly(residual.sugar,10)
                +poly(density,10)+poly(chlorides,10)+poly(volatile.acidity,10)
                +poly(fixed.acidity,10)+poly(total.sulfur.dioxide,10)
                +poly(sulphates,10)+ poly(free.sulfur.dioxide,10)+poly(citric.acid,10), data =
winetrainDataSet)
winemodel6
winetrainSetPrediction6 = predict(winemodel6, winetrainDataSet)
table(winetrainSetPrediction6$class, winetrainDataSet$quality_dummy)
mean(winetrainSetPrediction6$class== winetrainDataSet$quality_dummy)
#running model for test data
wineTestSetprediction6 = predict(winemodel6,winetestDataSet )
table(wineTestSetprediction6$class , winetestDataSet$quality_dummy)
mean(wineTestSetprediction6$class == winetestDataSet$quality_dummy)
#0.84

#creating seventh model
winemodel7 = lda(quality_dummy~poly(alcohol,8)+pH+poly(residual.sugar,4)
                +density+poly(chlorides,6)
                +poly(volatile.acidity,8)+fixed.acidity+total.sulfur.dioxide
                +sulphates, data = winetrainDataSet)
winemodel7
winetrainSetPrediction7 = predict(winemodel7, winetrainDataSet)
table(winetrainSetPrediction7$class, winetrainDataSet$quality_dummy)
mean(winetrainSetPrediction7$class== winetrainDataSet$quality_dummy)
#running model for test data
wineTestSetprediction7 = predict(winemodel7,winetestDataSet )

```

```

table(wineTestSetprediction7$class , winetestDataSet$quality_dummy)
mean(wineTestSetprediction7$class == winetestDataSet$quality_dummy)
#0.84

#creating conclusion graph to visualize the error.
attach(cleanwinedata)
model2D = lda(quality_dummy~poly(alkohol)+pH+residual.sugar+density+chlorides+
              volatile.acidity+fixed.acidity+total.sulfur.dioxide
              +sulphates+ free.sulfur.dioxide+citric.acid)
guess2D = predict(model2D, cleanwinedata)
winewith2Dguesses = mutate(cleanwinedata, guess=guess2D$class,
                             isError = ifelse(guess==quality_dummy, as.character(guess),
                                                "error"))
View(winewith2Dguesses)

ggplot(data= winewith2Dguesses )+geom_bar( aes(x= quality_dummy, fill=isError))+
  ylab("Number of wine")+
  xlab("Grouped quality of wine")+
  labs(color = "outcomes")+
  ggtitle("visualising error percentage in classifying wines")

ggplot(data=winewith2Dguesses)+
  geom_bar(aes(x= quality_dummy,
               fill=as.factor(quality_dummy)))+scale_fill_manual(values = c( "light green",
"orange"))+
  ylab("Number of wine")+
  xlab("Grouped quality of wine")+
  labs(color = "outcomes")+
  ggtitle("visualising actual number of wines classified as group")

```

```
#cross validation for changed quality

wineMix = slice(cleanwinedata, sample(1:n()))

summary(cleanwinedata)

summary(wineMix)

id = seq(1, 1143, by=1)

wineRando = mutate(wineMix, id)

k = 5

numRows = nrow(cleanwinedata)

train = filter(wineRando, id <= 4*numRows/k)

test = anti_join(wineRando, train, by="id")

View(train)

View(test)

library(MASS)

attach(cleanwinedata)

k = 5
```

```

numRows = nrow(cleanwinedata)

errors_1 = rep(0, k)

totalError = 0

for(i in 1:k){

  test = filter(wineRando, id >= (i-1)*numRows/k+1 & id <= i*numRows/k)

  train = anti_join(wineRando, test, by="id")

  model = lda(quality_dummy~alcohol+pH+residual.sugar+density, train)

  modelGuesses = predict(model, test)

  errors_1[i] = 1-mean(modelGuesses$class == test$quality_dummy)

  totalError = errors_1[i]+totalError

}

errors_1

totalError

avgerror=totalError/k

avgerror

```

```

#second model
errors_2 = rep(0, k)

totalError = 0

for(i in 1:k){

  test = filter(wineRando, id >= (i-1)*numRows/k+1 & id <= i*numRows/k)

  train = anti_join(wineRando, test, by="id")

  model = lda(quality_dummy~alcohol+pH+residual.sugar+density+chlorides, train)

  modelGuesses = predict(model, test)

  errors_2[i] = 1-mean(modelGuesses$class == test$quality_dummy)

  totalError = errors_2[i]+totalError

}

errors_2

totalError

avgerror=totalError/k

avgerror

#third model

```

```

errors_3 = rep(0, k)

totalError = 0

for(i in 1:k){

  test = filter(wineRando, id >= (i-1)*numRows/k+1 & id <= i*numRows/k)

  train = anti_join(wineRando, test, by="id")

  model = lda(quality_dummy~alcohol+pH+residual.sugar+density+chlorides+
              volatile.acidity, train)

  modelGuesses = predict(model, test)

  errors_3[i] = 1-mean(modelGuesses$class == test$quality_dummy)

  totalError = errors_3[i]+totalError

}

errors_3

totalError

avgerror=totalError/k

avgerror

```



```

#fourth model

errors_4 = rep(0, k)

totalError = 0

for(i in 1:k){

  test = filter(wineRando, id >= (i-1)*numRows/k+1 & id <= i*numRows/k)

  train = anti_join(wineRando, test, by="id")

  model = lda(quality_dummy~alcohol+pH+residual.sugar+density+chlorides+
              volatile.acidity+fixed.acidity+total.sulfur.dioxide, train)

  modelGuesses = predict(model, test)

  errors_4[i] = 1-mean(modelGuesses$class == test$quality_dummy)

  totalError = errors_4[i]+totalError

}

errors_4

totalError

avgerror=totalError/k

avgerror

```

```

#fifth model

errors_5 = rep(0, k)

totalError = 0

for(i in 1:k){

  test = filter(wineRando, id >= (i-1)*numRows/k+1 & id <= i*numRows/k)

  train = anti_join(wineRando, test, by="id")

  model = lda(quality_dummy~alcohol+pH+residual.sugar+density+chlorides+
              volatile.acidity+fixed.acidity+total.sulfur.dioxide
              +sulphates, train)

  modelGuesses = predict(model, test)

  errors_5[i] = 1-mean(modelGuesses$class == test$quality_dummy)

  totalError = errors_5[i]+totalError

}

errors_5

totalError

avgerror=totalError/k

```

```
avgerror
```

```
#sixth model
```

```
errors_6 = rep(0, k)
```

```
totalError = 0
```

```
for(i in 1:k){
```

```
  test = filter(wineRando, id >= (i-1)*numRows/k+1 & id <= i*numRows/k)
```

```
  train = anti_join(wineRando, test, by="id")
```

```
  model =  
lda(quality_dummy~poly(alcohol,10)+poly(pH,10)+poly(residual.sugar,10)+poly(density,10)  
+poly(chlorides,10)+poly(volatile.acidity,10)  
      +poly(fixed.acidity,10)+poly(total.sulfur.dioxide,10)+poly(sulphates,10)+  
poly(free.sulfur.dioxide,10)+poly(citric.acid,10)  
      , train)
```

```
  modelGuesses = predict(model, test)
```

```
  errors_6[i] = 1-mean(modelGuesses$class == test$quality_dummy)
```

```
  totalError = errors_6[i]+totalError
```

```
}
```

```
errors_6
```

```

totalError

avgerror=totalError/k

avgerror

#seventh model

errors_7 = rep(0, k)

totalError = 0

for(i in 1:k){

  test = filter(wineRando, id >= (i-1)*numRows/k+1 & id <= i*numRows/k)

  train = anti_join(wineRando, test, by="id")

  model =
lda(quality_dummy~poly(alcohol,8)+pH+poly(residual.sugar,4)+density+poly(chlorides,6)+
poly(volatile.acidity,8)+fixed.acidity+total.sulfur.dioxide
    +sulphates
    , train)

  modelGuesses = predict(model, test)

  errors_7[i] = 1-mean(modelGuesses$class == test$quality_dummy)

  totalError = errors_7[i]+totalError

```

```
}
```

```
errors_7
```

```
totalError
```

```
avgerror=totalError/k
```

```
avgerror
```

```
errors_1
```

```
errors_2
```

```
errors_3
```

```
errors_4
```

```
errors_5
```

```
errors_6
```

```
errors_7
```

```
avgE=rep(0,7)
```

```
avgE
```

```
for(i in 1:k){
```

avgE[1]=errors1[i]+avgE[1]

avgE[2]=errors2[i]+avgE[2]

avgE[3]=errors3[i]+avgE[3]

avgE[4]=errors4[i]+avgE[4]

avgE[5]=errors5[i]+avgE[5]

avgE[6]=errors6[i]+avgE[6]

avgE[7]=errors7[i]+avgE[7]

}

avgE[1]

avgE[2]

avgE[3]

avgE[4]

avgE[5]

avgE[6]

avgE[7]

```
se=rep(0,k)
```

```
se[1]=sqrt(var(errors1)/k)
```

```
se[2]=sqrt(var(errors2)/k)
```

```
se[3]=sqrt(var(errors3)/k)
```

```
se[4]=sqrt(var(errors4)/k)
```

```
se[5]=sqrt(var(errors5)/k)
```

```
se[6]=sqrt(var(errors6)/k)
```

```
se[7]=sqrt(var(errors7)/k)
```

```
mn=seq(1,7, by=1)
```

```
length(avgE)
```

```
length(se)
```

```
cross_validation2 = data.frame(avgE, se)
```

```
View(cross_validation2)
```

```
#plotting data
```

```

library(tidyverse)

ggplot(cross_validation2, aes(x=mn,y=avgE))+

  geom_line()+

  geom_point()+

  geom_errorbar(aes(ymin=avgE-se, ymax=avgE+se))+

  xlab("Model Complexity")+

  ylab("Classification Error")+

  ggtitle("Classification Error vs Model Complexity ")

#clustering:
#1D dataset
Selectivewine = dplyr::select(cleanwinedata, fixed.acidity)
m1 = kmeans(Selectivewine, 4, nstart=100)
m1$size
#histogram for fixed acidity
#k value is 3
m1 = kmeans(Selectivewine, 3, nstart=10)
m1$size

ggplot(data = Selectivewine, aes(x = fixed.acidity, fill = factor(m1$cluster)))+
  geom_histogram(stat = "count")+xlab("Fixed acidity Level")+
  ylab("Wine Count")+ labs(colour = "Cluster Number")+

```



```
ggtitle("Histogram on Fixed Acidity")+scale_colour_discrete(name = "Fixed Acidity  
Clusters",
```

```
labels = c("less acidic", "moderately acidic"  
           , "highly acidic"))
```

```
#2 dimensional
```

```
#k3 pH~alcohol
```

```
winescale = scale(cleanwinedata)
```

```
winescale = data.frame(winescale)
```

```
Selectivewine1 = dplyr::select(winescale, alcohol,pH)
```

```
M2 = kmeans(Selectivewine1, 3, nstart = 10)
```

```
ggplot(data = Selectivewine1, aes(x=alcohol, y = pH, colour = factor(M2$cluster)))+
```

```
geom_point()+
```

```
xlab("Alcohol level")+
```

```
ylab("pH Value")+
```

```
labs(colour = "heart rate range clusters")+
```

```
ggtitle("Relationship between pH~ Alcohol")+
```

```
scale_colour_discrete(name = "pH & alcohol",
```

```
labels = c("cluster 1", "cluster 2"  
           , "cluster 3"))
```

```
#clustering using 2 dimensional
```

```
#density~residual sugar
```

```
#k3
```

```
attach(winescale)
```

```
Selectivewine2 = dplyr::select(winescale, density, residual.sugar)
```

```
M3 = kmeans(Selectivewine2, 3, nstart = 10)
```

```
M3$centers
```

```
ggplot(data = Selectivewine2, aes(x = residual.sugar, y = density,
```

```
colour = factor(M3$cluster)))+
```

```

geom_point()+
xlab("Residual Sugar Level")+
ylab("Density of the wine")+
labs(colour = "heart rate range clusters")+
ggtitle("Relationship between Residual Sugar~ Density")+
scale_colour_discrete(name = "Residual Sugar & Density ",
                      labels = c("cluster 1","cluster 2","cluster 3"
                                ,"cluster 4"))

```

```

#clustering using 2 dimensional
#ph~ chlorides
#k3
attach(winescale)
Selectivewine3 = dplyr::select(winescale, pH,chlorides)
M3 = kmeans(Selectivewine3, 3, nstart = 10)
M3$centers
ggplot(data = Selectivewine3, aes(x = pH, y = chlorides,
                                colour = factor(M3$cluster)))+
geom_point()+
xlab("pH Value")+
ylab("CHLORIDE LEVELS")+
labs(colour = "heart rate range clusters")+
ggtitle("Relationship between pH~ Chlorides")+
scale_colour_discrete(name = "pH~ Chlorides Clusters ",
                      labels = c("cluster 1","cluster 2","cluster 3"
                                ,"cluster 4"))

```

```

#####PLOTING GRAPHS FOR DATA
VISUALISATION#####

```

```

#data visualisation
attach(cleanwinedata)

```

```

view(cleanwinedata)

ggplot(data= cleanwinedata)+
  geom_histogram(aes(x=fixed.acidity),
                 position = position_dodge2(padding = .1,
                                             preserve = "single"))+
  ylab("Wine Count")+
  xlab("Fixed acidity Level")+
  ggtitle("Histogram on Fixed Acidity ")

```

#quality vs alcohol level

```

ggplot(data = rawwinedata)+
  geom_point(mapping = aes (y = alcohol, x = citric.acid, color = (citric.acid)))+
  ylab("Alcohol level in of wine")+
  xlab("Citric acid level in wine")+
  ggtitle("Relationship between Citric acid and Alcohol level")

```

#point graph for chlorides and pH

```

ggplot(data = rawwinedata)+
  geom_point(mapping = aes (y = chlorides, x = pH))+
  xlab("pH of wine")+
  ylab("Chloride level in wine")+
  ggtitle("Relationship between pH and Chloride level")

```

#point graph for density and residual sugar

```

ggplot(data = rawwinedata)+
  geom_point(mapping = aes (x = density, y = residual.sugar))+
  ylab("Residual Sugar level of wine")+
  xlab("Density of wine")+
  ggtitle("Relationship between pH and Chloride level")

```

```
#boxplot quality and pH
ggplot(data = rawwinedata)+
  geom_boxplot(mapping = aes (x = as.factor(quality), y = pH, color =as.factor(quality)))+
  ylab("pH of wine")+
  xlab("Quality level of wine")+
  ggtitle("Relationshhip between pH level and quality")+
  scale_colour_discrete(name = "quality of wine")
```

```
#boxplot for quality and alcohol
ggplot(data = rawwinedata)+
  geom_boxplot(mapping = aes (x = as.factor(quality), y = alcohol, color
=as.factor(quality)))+
  ylab("Alcohol level in wine")+
  xlab("quality of wine")+
  ggtitle("Relationshhip between quality and alcohol")+
  scale_colour_discrete(name = "quality of wine")
```